

<div>Estimation and Planning Mistakes</div> <ul style="list-style-type: none"><li>● <b>Underestimating complexity, cost, and/or schedule</b><ul style="list-style-type: none"><li>– Use historical data and expert judgment to estimate accurately.</li></ul></li><li>● <b>Abandoning planning under pressure</b><ul style="list-style-type: none"><li>– Stick to planning to avoid chaotic code-and-fix mode.</li></ul></li><li>● <b>Overly aggressive schedules</b><ul style="list-style-type: none"><li>– Set realistic schedules based on historical data and project complexity.</li></ul></li><li>● <b>Wasting time in the “fuzzy front end”</b><ul style="list-style-type: none"><li>– Streamline the approval and budgeting process.</li></ul></li></ul>	<div>Quality and Risk Management Mistakes</div> <ul style="list-style-type: none"><li>● <b>Poor quality workmanship</b><ul style="list-style-type: none"><li>– Implement quality assurance processes and conduct regular code reviews.</li></ul></li><li>● <b>No risk management</b><ul style="list-style-type: none"><li>– Identify risks early and develop mitigation plans.</li></ul></li><li>● <b>Ignoring system performance requirements</b><ul style="list-style-type: none"><li>– Define and monitor performance requirements throughout the project.</li></ul></li><li>● <b>Poorly planned/managed transitions</b><ul style="list-style-type: none"><li>– Develop detailed transition plans and involve all relevant parties.</li></ul></li></ul>	<div>Scrum Roles</div> <ul style="list-style-type: none"><li>● Product Owner:<ul style="list-style-type: none"><li>– Maximizes product value.</li><li>– Develops and communicates Product Goal.</li><li>– Creates and prioritizes Product Backlog.</li><li>– Ensures transparency and understanding of Backlog.</li><li>– One person, not a committee, with leadership role.</li></ul></li><li>● Scrum Master:<ul style="list-style-type: none"><li>– Facilitates Scrum process, resolves impediments.</li><li>– Creates self-organization environment.</li><li>– Captures empirical data, shields team from distractions.</li><li>– Enforces timeboxes, keeps artifacts visible.</li><li>– Promotes improved practices, has leadership role.</li></ul></li><li>● Development Team:<ul style="list-style-type: none"><li>– Develops product, self-organizing, cross-functional.</li><li>– No titles, no sub-teams, no specialized roles.</li><li>– Long-term, full-time membership, 7 ± 2 members.</li></ul></li></ul>	<div>Scrum Artifacts</div> <ul style="list-style-type: none"><li>● <b>Product Backlog</b><ul style="list-style-type: none"><li>– Prioritized list of features.</li><li>– Updated regularly.</li><li>– Visible to all stakeholders.</li><li>– Owned by Product Owner.</li></ul></li><li>● <b>Sprint Backlog</b><ul style="list-style-type: none"><li>– List of tasks for current Sprint.</li><li>– Owned by Development Team.</li><li>– Updated daily.</li><li>– Created during Sprint Planning Meeting.</li><li>– Decomposed from Product Backlog.</li></ul></li><li>● <b>Burndown Charts</b><ul style="list-style-type: none"><li>– Graphical representation of work remaining.</li><li>– Updated daily.</li><li>– Shows progress towards Sprint Goal.</li><li>– Helps identify issues early.</li><li>– Used to forecast project completion.</li></ul></li></ul>	<div>Agile Sweet Spots</div> <ul style="list-style-type: none"><li>● Dedicated developers.</li><li>● Experienced developers.</li><li>● Small co-located team.</li><li>● Tools for testing and configuration management.</li><li>● Easy user access.</li><li>● Short increments and frequent delivery.</li></ul>
<div>Communication and Stakeholder Engagement Mistakes</div> <ul style="list-style-type: none"><li>● <b>Poor communication</b><ul style="list-style-type: none"><li>– Hold regular meetings and ensure clear documentation.</li></ul></li><li>● <b>Not engaging stakeholders</b><ul style="list-style-type: none"><li>– Include stakeholders in planning and review sessions.</li></ul></li><li>● <b>Insufficient user input</b><ul style="list-style-type: none"><li>– Ensure active involvement of end-users throughout the project.</li></ul></li></ul>	<div>Agile Manifesto</div> <ul style="list-style-type: none"><li>● Individuals and interactions over processes and tools.</li><li>● Working software over comprehensive documentation.</li><li>● Customer collaboration over contract negotiation.</li><li>● Responding to change over following a plan.</li></ul>		<div>Accidental vs Essential Complexity</div> <ul style="list-style-type: none"><li>● Essential complexity: - Inherently difficult problems with no known solution.</li><li>● Necessary accidental complexity: - Example: project management.</li><li>● Unnecessary accidental complexity: - Waste, Lean, MEI (minimum essential information).</li></ul>	
<div>Project Management Mistakes</div> <ul style="list-style-type: none"><li>● <b>Lack of oversight/poor project management</b><ul style="list-style-type: none"><li>– Appoint experienced project managers and conduct regular reviews.</li></ul></li><li>● <b>Adding developers to a late project</b><ul style="list-style-type: none"><li>– Avoid adding developers late in the project to prevent further delays.</li></ul></li></ul>	<div>Agile Principles</div> <ul style="list-style-type: none"><li>● Satisfy the customer with continuous delivery.</li><li>● Welcome changing requirements.</li><li>● Frequent delivery of working software.</li><li>● Daily collaboration between business and developers.</li><li>● Build projects around motivated individuals.</li><li>● Face-to-face conversation for communication.</li><li>● Working software as progress measure.</li><li>● Promote sustainable development.</li><li>● Continuous attention to technical excellence.</li><li>● Simplicity is essential.</li><li>● Best architectures emerge from self-organizing teams.</li><li>● Regular reflection and adjustment.</li></ul>	<div>Scrum Events</div> <ul style="list-style-type: none"><li>● Sprints</li><li>● Sprint Planning Meeting</li><li>● Daily Scrum Meeting</li><li>● Sprint Review Meeting</li><li>● Sprint Retrospective</li></ul>	<div>Best/Good/Recommended Practices</div> <ul style="list-style-type: none"><li>● “Best Practice”: - Consistently improves productivity, cost, schedule, quality, user satisfaction, predictability.</li><li>● Best Practices (Glass, 2004): - Development teams repeat mistakes. - Best practice documents regurgitate textbook material. - Growing field’s wisdom not increasing.</li></ul>	